Hindawi
Mathematical Problems in Engineering
Volume 2023, Article ID 2005465, 12 pages
https://doi.org/10.1155/2023/2005465

Research Article

A Two-Archive Algorithm with Interactive Mechanism for Many-Objective Optimization

Shuren Liu,[1,2] **Shuping Li**,[1,2] **Changchun Li**,[1,2] **Changning Cai**,[1,2] **and Guojian Cheng**[3]

[1]*Northwest Branch of PetroChina Research Institute of Petroleum Exploration and Development, Lanzhou 730020, China*
[2]*Key Laboratory of Internet of Things, China National Petroleum Corporation, CNPC, Lanzhou 730020, China*
[3]*Xi'an Shiyou University, Xi'an 710065, Shaanxi Province, China*

Correspondence should be addressed to Shuren Liu; liushr@petrochina.com.cn

Received 12 March 2023; Revised 20 June 2023; Accepted 29 November 2023; Published 19 December 2023

Academic Editor: Shangce Gao

This paper proposes an interactive two-archive method to solve many-objective optimization problems. Two updating strategies based on the aggregation-based framework are presented and incorporated into a two-archive framework. In addition, we further extend this method by introducing an interactive mechanism in which evolutionary information is passed from the diversity archive to the convergence archive. Given the requirement to balance convergence and diversity, a mating selection method is proposed to regulate the evolutionary speed of these two archives collaboratively. The proposed algorithm has been tested extensively on several problems with different peer algorithms to validate its effectiveness. The results show that the proposed method can outperform several state-of-the-art evolutionary algorithms for handling many-objective optimization.

1. Introduction

Multi-objective optimization problems (MOPs), which refer to the task of optimization with more than one objective, commonly exist in real-world applications. e.g., engineering design problems [1], software engineering problems [2], and water management [3]. Generally, a continuous MOP can be defined as follows:

$$\begin{aligned} &\text{minimize} \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\ &s.t. \quad \mathbf{x} \in \Omega \end{aligned}, \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the decision vector, Ω is the decision space, and $\mathbf{F}: \Omega \to \mathbb{R}^m$ is the objective function vector, mapping from the n-dimensional decision space Ω to the objective space $\mathbb{R}^m$. As a specific category of MOPs, many-objective optimization problems (MaOPs) refer to MOPs with more than three objectives ($m \geq 4$). The reason for separating this single category is that a large number of multi-objective evolutionary algorithms (MOEAs) encounter substantial difficulties when solving MaOPs. For example, two representative Pareto-based methods, NSGA-II [4], and SPEA2 [5] have shown to be very effective in handling MOPs but degraded drastically in addressing MaOPs due to dominance resistance [6]. Specifically, the proportion of nondominated solutions increases significantly in higher objective space, and the strictly defined dominance relationship cannot discriminate between these solutions.

As the weakest preferred structure for the decision-makers, Pareto dominance can be defined as follows: a vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)^T$ is said to dominate another vector $\mathbf{y} = (y_1, y_2, \ldots, y_m)^T$, represents as $\mathbf{x} \prec \mathbf{y}$, if $\forall i \in \{1, 2, \ldots, m\}$, $x_i \leq y_i$ and $\exists j \in \{1, 2, \ldots, m\}, x_j < y_j$. In order to address the issues caused by dominance resistance, there are a number of methods to improve the Pareto-based method for MaOPs. The most intuitive attempt is to relax the original dominant relationship, making solutions more comparable in a higher objective space. Some studies, such as ϵ-dominance [7], fuzzy-dominance [8], and θ-dominance [9], were proposed. By enlarging the dominated area of a solution, these new dominances can increase the convergence ability to some extent. Nevertheless, setting a proper value for the parameters

in these methods is not an easy task since it varies over problems and relies on the number of objectives.

Objective reduction is a direct way to handle dominance resistance by converting a MaOP into a MOP. For some real-life MaOPs, their objectives can be highly correlated. Thus, the basic idea behind these methods is to keep the essential objectives and discard the redundant ones. After object reduction, if the remaining number of objectives is lower than four, any state-of-the-art MOEA can be directly introduced. Various machine learning algorithms, such as principal component analysis (PCA), maximum variance unfolding (MVU) [10], and feature selection [11], can be considered tools for objective reduction.

Additionally, even MOEA can be used to remove redundant objectives by considering different kinds of errors simultaneously [12]. The advantage of the objective-reduction-based method is that we intend to do something other than any extra ways to promote the scalability of existing MOEAs. Yet, these reduction strategies may only be valid when more redundant objectives exist.

Indicator-based methods combine convergence and diversity performance into a single indicator as the selection criteria in environmental selection. Theoretically, they can avoid the dominance resistance phenomenon, but in practise, it is doubtful that they will be able to maintain population coverage and diversity effectively when dealing with MaOPs. The hypervolume estimation (HypE) [13] and the indicator-based evolutionary algorithm (IBEA) [14] can be considered as two representatives in this category. Because of its strong theoretical qualities, the HypE are commonly utilized in solving MaOPs, but their computing complexity grows exponentially as the number of objectives grows. The $I_{\epsilon+}$ indicator in IBEA and its variant [15] are gaining popularity in MaOPs as $I_{\epsilon+}$ does not require a reference point or other prior knowledge of the real PF and has a relatively low-computational complexity. However, $I_{\epsilon+}$ only exhibits good performance in termes of convergence, whereas diversity tends to be poor; combing both convergence and diversity performance into a single indicator is not a easy task.

Another promising method for MaOPs is to decompose a MOP into several subproblems by using a series of aggregation functions, and then solve these subproblems simultaneously. These methods can be referred to as decomposition-based or aggregation-based algorithms. The MOEA based on decomposition (MOEA/D) can be considered the most representative of this class of methods [16]. Although the MOEA/D is not originally proposed for MaOPs, it is still competitive to handle in the many-objective scenario. So far, many variants based on MOEA/D have been proposed from different aspects. e.g., adaptively adjusting weight vectors [17], exploiting the perpendicular distance from solution to weight vectors [18], enhancing diversity by a hybrid method [19].

The basic principle for designing an MOEA is to find a set of nondominated solutions that are distributively close to the Pareto front (PF), reflecting two goals of the search process: convergence and diversity. Thus, some literature begins to separate these two basic goals into different subpopulations; this strategy begins to focus on the structure of selection instead of a specific selection procedure. Two-archive based method can be considered as a representative method following this idea [20]. As one of hybrid methods [21], typically, in two-archive-based method, the convergence archive (CA) targets the solutions with good convergence, and the diversity archive searches for the solutions with good diversity. Both of these evolutionary processes have occurred separately. The original two-archive algorithm is unsuitable for MaOPs, since, in the absence of any restrictions, the number of nondominated solutions in CA would increase dramatically, failing to provide sufficient selection pressure [20]. In Two Archive 2 (Two_Arch2), the structure and survival strategies were updated significantly: the size of two archives are fixed, and dominance-based methods are not directly utilized to avoid dominance resistance [22]. Cai et al. [23] proposed a new two-archive method based on an aggregation-based framework that combines the benefits of two archive and aggregation-based methods. Besides, as the demands of convergence and diversity are decoupled in two different archives, two-archive techniques are better suited for tackling multimodel MOPs [24] and constrained MOPs [25].

Although the two-archive methods have been extensively used for MaOPs, they still suffer many deficiencies when applied to the other cases. First, the two archives evolve their populations according to their distinct standards, and there is no interaction between these two archives. Second, different problems have different characteristics; some MOPs converge quickly, while others may necessitate an emphasis on diversity. As a result, the two-archive method with separate yet fixed evolutionary strategies fails to balance convergence and diversity in some cases.

To address such issues, we propose an interactive two-archive method for aggregation-based MOEA, termed iTwoArch. The main contributions of this work are summarized as follows:

(i) In iTwoArch, unlike its predecessor, replacement information on subproblems in DA can be transmitted to CA, thereby avoiding redundant searches to some extent.

(ii) A mating selection is carefully designed according to the evolutionary speed of these two archives, which effectively enhances the capability of balancing convergence and diversity in iTwoArch.

(iii) Different kinds of test problems are used to verify the performance of the proposed method, compared with several state-of-the-art peer algorithms.

The remainder of this paper is organized as follows: the proposed algorithm is described in detail in the next section. The third section presents the experimental design, test problems, and performance indicators for investigating the performance of the proposed iTwoArch, followed by experimental results and relevant discussions. Finally, the conclusion and some possible future research directions are given in the last section.

1: $\Lambda \leftarrow$ weightVectorsGenerator()
2: $(P_{CA}, P_{DA}) \leftarrow$ initializePopulation()
3: $\mathbf{z}^* \leftarrow$ initializeIdealPoint(P_{CA}, P_{DA})
4: $B \leftarrow$ initializeNeighborhood()
5: $\epsilon_{CA} \leftarrow 0, \epsilon_{DA} \leftarrow 0$
6: **while** termination criterion is not fulfilled **do**
7: **for** $i \leftarrow 1$ to N **do**
8: $(\mathbf{x}_i, \mathbf{y}_i) \leftarrow$ matingSelection($B, \epsilon_{CA}, \epsilon_{DA}$)
9: $\mathbf{s} \leftarrow$ crossover($\mathbf{x}_i, \mathbf{y}_i$)
10: $\mathbf{s} \leftarrow$ mutation($\mathbf{s}$)
11: updateIdealPoint($\mathbf{s}, \mathbf{z}^*$)
12: $k' \leftarrow$ updateDA($\mathbf{s}, \mathbf{z}^*, \Lambda, P_{DA}, \epsilon_{DA}$)
13: updateCA($\mathbf{s}, \mathbf{z}^*, \Lambda, P_{CA}, \epsilon_{CA}, k'$)
14: **end for**
15: $\epsilon_{CA} \leftarrow 0, \epsilon_{DA} \leftarrow 0$
16: **end while**
17: **return** all the non-dominated solutions in P_{DA}

Algorithm 1: Framework of the proposed algorithm.

Input: offspring solution $\mathbf{s}$, ideal point set $\mathbf{z}^*$, weight vector set Λ, diversity archive P_{DA}, count of replacement in current generation ϵ_{DA}
Output: index of matched sub-problem i
1: $i \leftarrow$ associate($\mathbf{s}, \mathbf{z}^*, \Lambda$)
2: **if** $\mathbf{s} \prec \mathbf{x_i}$ **then**
3: $P_{DA} \leftarrow P_{DA} \setminus \{\mathbf{x}_i\}$, $P_{DA} \leftarrow P_{DA} \cup \{\mathbf{s}\}, \epsilon_{DA} \leftarrow \epsilon_{DA} + 1$
4: **else**
5: **if** $\mathbf{s}$ and $\mathbf{x}_i$ are nondominated with each other **then**
6: **if** $d^{\perp}(\mathbf{s}, \lambda_i, \mathbf{z}^*) < d^{\perp}(\mathbf{x}_i, \lambda_i, \mathbf{z}^*)$ **then**
7: $P_{DA} \leftarrow P_{DA} \setminus \{\mathbf{x}_i\}$, $P_{DA} \leftarrow P_{DA} \cup \{\mathbf{s}\}, \epsilon_{DA} \leftarrow \epsilon_{DA} + 1$
8: **end if**
9: **else**
10: $i \leftarrow -1$
11: **end if**
12: **end if**
13: **return** i

Algorithm 2: Update procedure for diversity archive.

2. The Proposed Method

2.1. Main Idea. Algorithm 1 gives the general framework of the proposed iTwoArch; like many other steady-state algorithms [16, 23]. First, a series of uniformly spread weight vectors are generated in advance. Das and Dennis [26] systematic approach is used, and the two-layer method is adopted for the number of objectives over seven [23]. Then, two populations are initialized separately. The ideal point set $\mathbf{z}^*$, which is defined as the best scalar value for each objective in the current state, is obtained from two populations (Step 3). Next, the neighborhood for each subproblem is determined in Step 4. After that, every vector in B stores the indexes of T closet weight vectors, i.e., $B(i) = \{i_i, i_2, \ldots, i_T\}$, where $\{\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{iT}\} \in \Lambda$ indicates T closet weight vectors to current λ_i. Steps 6–16 are iterated until the termination criterion is fulfilled. Specifically, in each iteration, two mating solutions are chosen according to the replacement status of two archives. Once parents are determined, simulated binary crossover (SBX) and polynomial mutation are executed sequentially to generate a new solution $\mathbf{s}$. Finally, $\mathbf{s}$ is used to update the ideal point set and two different populations.

After presenting the framework of the proposed algorithm, we would like to discuss the main similarities and differences between the improved version and the original TwoArchA [23]. For the similarities, both introduce a set of weight vectors to guide the search behavior in two archives separately. Besides, both of them use the steady-state method to update two archives, which means that once an offspring $\mathbf{s}$ is produced, it directly goes through twin archives to determine whether or not to replace a specific solution. However, compared with its predecessor, the characteristic procedure of the proposed algorithm mainly lies in two updating steps and mating selection. In our design, CA and DA's evolution is based not only on different aims but also on interaction. The results of updating DA are considered necessary for the evolution of CA. Additionally, the mating selection in this study intends to address two issues: (1) to prevent two distant solutions from being combined; (2) to balance the convergence and diversity according to the state of the two archives. In the following subsection, we would like to present the two updated mechanisms and mating selection in detail.

2.2. Diversity Archive. The update mechanism of the DA is presented in detail in Algorithm 2. As we know, the main goal of DA is to select well-diversified solutions. The solutions in this archive should be distributed in the PF with large diversified degrees. For that, some characteristic procedures are designed.

Specifically, when the new solution $\mathbf{s}$ is created, we first find the most suitable weight λ for $\mathbf{s}$ via association (Step 1 of Algorithm 2). The procedure of association is presented in the Algorithm 3. For each weight vectors $\lambda_i \in \Lambda$, we compute perpendicular distance from $\mathbf{s}$ to λ_i as follows:

$$d_i^{\perp} = \left\| \widehat{\mathbf{f}}(\mathbf{s}) - \lambda_i \right\| \widehat{\mathbf{f}}(\mathbf{s}) \cdot \lambda_i \| / \| \lambda_i^2 \| \Big\|, \tag{2}$$

where $\widehat{\mathbf{f}}(\mathbf{s})$ is translated objectives: $\widehat{f}_j(\mathbf{s}) = f_j(\mathbf{s}) - z_j^*$ ($z_j^* \in \mathbf{z}^*$, $j = 1, \ldots, m$). After the appropriate weight λ_i is determined, the original solution associated to this weight $\mathbf{x}_i$ competes with $\mathbf{s}$, and the operation of replacement happens in two cases:

(1) $\mathbf{s}$ dominates $\mathbf{x}_i$ (Step 3 in Algorithm 2).
(2) if $\mathbf{s}$ and $\mathbf{x}_i$ are nondominated with each, but $\mathbf{s}$ has lower perpendicular distance to λ_i (Step 7 in Algorithm 2).

Input: offspring solution $\mathbf{s}$, ideal point set $\mathbf{z}^*$, weight vector set Λ
Output: index of matched sub-problem i
1: **for** each $\lambda_i \in \Lambda$ **do**
2: Compute the perpendicular distance $d^{\perp}(\mathbf{s}, \lambda_i, \mathbf{z}^*)$
3: **end for**
4: Assign $\widehat{\lambda}_i = \lambda_i : argmin_{\lambda_i \in \Lambda} d^{\perp}(\mathbf{s}, \lambda_i, \mathbf{z}^*)$
5: **return** i

Algorithm 3: Association.

Input: offspring solution $\mathbf{s}$, ideal point set $\mathbf{z}^*$, weight vector set Λ, convergence archive P_{CA}, count of replacement in current generation ϵ_{CA}, k' the index of sub-problem
1: $E \leftarrow B(k')$
2: **for** each $i \in E$ **do**
3: **if** $\mathbf{s} \prec \mathbf{y}_i$ **then**
4: $P_{CA} \leftarrow P_{CA} \setminus \{\mathbf{y}_i\}, P_{CA} \leftarrow P_{CA} \cup \{\mathbf{s}\}, \epsilon_{CA} \leftarrow \epsilon_{CA} + 1$
5: **else**
6: **if** $\mathscr{F}_i(\mathbf{s}) < \mathscr{F}_i(\mathbf{y}_i)$ **then**
7: $P_{CA} \leftarrow P_{CA} \setminus \{\mathbf{y}_i\}, \quad P_{CA} \leftarrow P_{CA} \cup \{\mathbf{s}\}, \epsilon_{CA} \leftarrow \epsilon_{CA} + 1$
8: **end if**
9: **end if**
10: **end for**
11: **return**

Algorithm 4: Update procedure for convergence archive.

The replacement count ϵ_{DA} increases once a replacement happens.

2.3. Convergence Archive. The main idea of the update mechanism in CA is to evaluate the convergence in the ith subproblem: for each solution, $\mathbf{y}_i$ in the neighborhood E, replace $\mathbf{y}_i$ if $\mathbf{s}$ is better than $\mathbf{y}_i$ in terms of convergence. The update procedure for convergence archive is shown in Algorithm 4. Similar to the mechanism of DA, we still use two standards to evaluate the ability of convergence. First, if $\mathbf{s}$ dominates $\mathbf{y}_i$, $\mathbf{s}$ can be considered as a substitute for $\mathbf{y}_i$. In addition to that, if $\mathbf{s}$ has a smaller aggregation function value, i.e., $\mathscr{F}_i(\mathbf{s}) < \mathscr{F}_i(\mathbf{y}_i)$, $\mathbf{s}$ can also be considered as a winner.

Like TwoArchA, we still use a modified version of the Tchebycheff function as the aggregation function in this study. A smaller Tchebycheff function value of a solution implies its better convergence performance. The aggregation function for j-th weight vector can be defined as follows:

$$\mathscr{F}_j(\mathbf{x}) = \max_{k=1}^{m} \left\{ \frac{1}{\lambda_{j,k}} \widehat{f}_k(\mathbf{x}) \right\}, \tag{3}$$

where $\widehat{f}_k(\mathbf{x})$ is the translated objective value for k-th objective value $\widehat{f}_k(\mathbf{x}) = f_k(\mathbf{x}) - z_k^*$.

Input: i index of sub-problem, δ probability of selection in neighbourhood, ϵ_{CA}, ϵ_{DA}, P_{CA}, P_{DA}
Output: the selected two solutions $\mathbf{x}, \mathbf{y}$
1: **if** rand() $< \delta$ **then**
2: $E \leftarrow B(i)$
3: **else**
4: $E \leftarrow \{1, 2, \ldots, N\}$
5: **end if**
6: Randomly select two indexes k_1 and k_2 from E
7: $\epsilon \leftarrow \frac{\epsilon_{DA}}{\epsilon_{DA} + \epsilon_{CA}}$
8: **if** rand() $< \epsilon$ **then**
9: $\mathbf{x} \leftarrow P_{CA}[k_1]$, $\mathbf{y} \leftarrow P_{CA}[k_2]$
10: **else**
11: $\mathbf{x} \leftarrow P_{CA}[k_1]$, $\mathbf{y} \leftarrow P_{DA}[k_2]$
12: **end if**
13: **return x, y**

Algorithm 5: Mating selection.

It is worth noting that the index of the best-matched subproblem k' is passed by the update procedure of DA, which means that we use perpendicular distance to confirm the most suitable subproblem for the replacement scheme in CA.

2.4. Mating Selection. Algorithm 5 provides the pseudocode of the whole mating selection. Commonly, mating selection aims at selecting better solutions for variation. In two-archive-based methods, the different solutions generally come from two archives independently, based on the assumption that combining them could generate a better solution by inheriting all abilities from their parents. Based on this idea, the mating selection in this work contains two stages.

First, taking advantage of the subregion defined by weight vectors, we restrict the solutions to be selected within some regions. This restriction may help alleviate issues in MaOPs, where recombining two distance solutions is not likely to generate good offspring. Additionally, to enhance the exploration ability, we also allow selection from the whole population rarely happens in the entire population (Step 4 in Algorithm 5).

The second stage is the collaborative process (Steps 7–12). Typically, two mating parents are picked up from CA and DA separately when two archives have a similar status. However, in some cases, the evolution of DA is faster than CA, which means there is an imbalance between convergence and diversity. Therefore, we need to control the speed of the evolutionary process in two archives. To make the collaboration work, there are two issues that need to be adequately attended to. First, estimating the state of two archives is necessary. In our design, we use replacement frequency as a criterion for state estimation. ϵ_{CA} and ϵ_{DA}, which count the replacement in CA and DA, respectively, can be directly considered as the indicators for the evolutionary state. Second, the control of the evolutionary process should be adequate. Here, we use a straightforward strategy: when DA has a higher proportion

of replacement, the mating selection is more likely carried in CA only (Step 9 in Algorithm 5).

2.5. Discussion. This subsection is devoted to discussing the main similarities and differences between the proposed method and its predecessors.

iTwoArch vs. TwoArch2 [22]: Despite the fact that they both use the two-archive frames to lead the solution set to the PF, the updating procedure in each archive is distinct. In TwoArch2, CA is maintained by an indicator-based method, whereas DA is maintained using a Pareto-based method, necessitating a new L_p-norm based diversity management strategy. For the proposed iTwoArch, both CA and DA are maintained by the decomposition-based selection method.

iTwoArch vs. TwoArchA+ [23]: both employ a steady-state decomposition-based approach to updating two archives, but their detailed maintenance methods are quite different. The updating of two archives is done independently in TwoArchA+, with no interaction. However, with the proposed iTwoArch, the updated solution information in the DA would be passed to the CA update process to better assist the selection. In addition, there is no specific mating selection designed for TwoArchA+, yet in our iTwoArch, the mating selection is based on the frequency of updates for two archives, keeping the two updating procedures more balanced.

3. Experimental Design

This section introduces test problems, performance metrics, algorithms in comparison, and parameter settings for the experimental studies.

3.1. Test Problems. To test the effectiveness of the proposed iTwoArchD, two well-known continuous benchmark suites, DTLZ1-4 [27] and WFG1-9 [28], are involved in our empirical studies. In particular, the number of objectives for each problem is set as $m \in \{3, 5, 8, 10, 15\}$. For the DTLZ problems, the number of decision variables is set to $n = m + k - 1$, where $k = 5$ for DTLZ1, and $k = 10$ for DTLZ2-4. As for WFG problems, we set the number of decision variables $n = 24$, and the position-related parameter is $m - 1$. For a fair comparison, we apply the same conditions to each test problem.

3.2. Parameter Settings. For a fair comparison, we utilize the same parameter settings with common parameters for the proposed iTwoArch and its rivals. First, in this research, we employ SBX and polynomial mutation to determine the parameters for the reproduction processes. The crossover and mutation probabilities are set to 1 and $1/n$, respectively. Besides, the the crossover and mutation distribution index has been set at 30 and 20, respectively. Following the suggestion of its original works, the neighborhood size for the decomposition-based method is set to 20, and the probability is set to 0.9. Furthermore, the population size is kept the same to ensure a fair comparison, and it is controlled by two parameters (H_1 and H_2) since the two-layer weight vector method is used in this works [29]. The detailed population size is summarized in Table 1.

Table 1: The population size.

No. of objectives	H_1	H_2	Population size
3	12	–	91
5	6	–	210
8	3	3	240
10	3	2	275
15	2	1	135

3.3. Algorithms in Comparisons. Here, we choose five state-of-the-art many-objective optimization algorithms to comprehensively study the performance of the proposed algorithm. These algorithms have covered different categories of MaOPs, including two decomposition-based algorithms (MOEA/D and MOEA/D-M2M), an algorithm that combines Pareto-based and decomposition-based selection together (NSGAIII), and two algorithms based on two archives (TwoArchA+ and TwoArch2). Some brief reviews and comments are listed as follows:

(i) *MOEA/D* [16]: MOEA/D can be considered a representative algorithm of the decomposition-based method. This study chooses the original MOEA/D with a modified version of the Tchebycheff function for a fair comparison.

(ii) MOEA/D-AM2M [30]: lacking prior knowledge of the PF, aggregation-based methods are hard to initialize an applicable set of weight vectors for a specific problem. To address this issue, MOEA/D-AM2M assumes that the current evolutionary population can be an approximation to the PF, and periodically resets the subregion setting and weight vectors. This practice of resetting makes MOEA/D-AM2M suitable for some degenerated MaOPs.

(iii) NSGAIII [29]: in NSGAIII, the maintenance of diversity is also aided by a series of weight vectors (or called reference lines in NSGAIII). Based on that, a niche-preservation operation was employed to identify the promising candidates. Overall, NSGAIII prefers some solutions that are nondominated but close to a set of well-distributed reference lines.

(iv) TwoArch2 [22]: TwoArch2 maintains two archives according to indicator-based selection and an L_p-norm-based diversity maintenance method. Different solutions are picked up from two archives containing different characteristics.

(v) TwoArchA+ [23]: a new two-archive method to deal with MaOPs. Similar to TwoArch2, the updating mechanism of CA and DA is based on different rules yet under the same decomposition-based framework. To further control the diversity, each subproblem's neighborhood has been extended to the whole population.

All algorithms, including the proposed iTwoArch, are implemented using Java and the jMetal framework [31], and all experiments are conducted on a Lenovo ThinkCentre computer with an Intel(R) Core i5-8400 (2.8 GHz) processor

Table 2: Performance comparison on DTLZ1-4 problems with respect to the average HV values.

Problem	Obj.	iTwoArch	TwoArchA+	TwoArch2	MOEAD	MOEADAM2M	NSGAIII
DTLZ1	3	0.973505	0.973558	**0.975170**	0.973651	0.970025†	0.972512†
	5	**0.998961**	0.988324†	0.997340†	0.996220†	0.998330†	0.998628†
	8	**0.999931**	0.999914†	0.999242†	0.999560†	0.999926	0.999604†
	10	**0.999999**	0.999775†	0.998868†	0.999957†	0.999988†	0.999731†
	15	**0.999997**	0.999991†	0.974723†	0.999227†	0.988791†	0.999710†
DTLZ2	3	0.926717	0.926697†	**0.929175**	0.926730	0.926001†	0.926643†
	5	**0.990526**	0.990386†	0.983421†	0.986239†	0.990349†	0.990434†
	8	**0.999321**	0.999302†	0.993856†	0.998597†	0.999246†	0.999259†
	10	**0.999918**	0.999901†	0.992931†	0.999772†	0.999875†	0.999876†
	15	0.999996	**0.999998**	0.981044†	0.999624†	0.999506†	0.999996
DTLZ3	3	0.978073	**0.997819**	0.972712	0.978151	0.976368†	0.977728†
	5	**0.998658**	0.960642	0.992559†	0.997989†	0.998381†	0.990501†
	8	0.999954	0.996681	0.817973†	0.999784†	**0.999956**	0.993984†
	10	0.999668	0.969624	0.000000†	0.999886	**0.999994**	0.946079
	15	0.999965	**0.999999**	0.691400†	0.999417†	0.997129†	0.999696†
DTLZ4	3	0.926715	0.909868†	**0.929180**	0.918331	0.926035†	0.926660†
	5	**0.990541**	0.990522†	0.985053†	0.985951†	0.990448†	0.990503†
	8	0.999370	0.999297	0.994405†	0.999350	**0.999420**	0.999355†
	10	0.999925	0.999913†	0.993294†	0.999916	**0.999932**	0.999919†
	15	**0.999999**	0.999995†	0.983069†	0.999992	0.999997†	0.999999

Note. "†" means that the result is significantly outperformed by iTwoArch.

and 64-GB RAM. Each algorithm is independently executed 30 times on each test instance, and the average values of the evaluation metrics are recorded. Furthermore, the Wilcoxon [32] signed-rank test at 5% significance level is performed on the metric values generated by two competing algorithms in order to examine the difference for statistical significance in different instances.

3.4. Performance Metrics. We used a hypervolume indicator (HV) as a performance indicator in the experiment. The good theoretical qualities of HV make it relatively fair to evaluate the performance of algorithms. And it can simultaneously measure an algorithm's convergence and diversity ability. The larger the HV value, the better the quality of obtained solutions to approximate the whole PF. Mathematically, the exact definition of HV is defined by Equaion (4) as follows:

$$\mathrm{HV}(S) = \{\bigcup \mathrm{volume}(v, \mathbf{r}) | v \in S, r \in \mathbf{z},\} \tag{4}$$

where $\mathbf{r} = \{r_1, \ldots, r_m\}^T$ is a reference point set which is dominated by all Pareto-optimal objective vectors. Thus, before calculating HV, a reference point is to be assigned. In our experiment, the setting of reference points for each test instance is the same as the experiments on TwoArchA [23], and the presented HV values in this paper are all normalized by dividing the corresponding $\prod_{i=1}^{m} r_i$. As for the problem with 15 objectives, the HV value is approximated by Monte Carlo simulation, and 10,000,000 points are sampled to ensure accuracy.

4. Empirical Result and Discussion

4.1. Performance Comparison on DTLZ Problems. For the DTLZ1 problem, it has a linear PF with a large number of optima, arousing a challenge for an algorithm to converge toward PF. According to the results in Table 2, iTwoArch shows better performance than the other algorithms in almost all instances. For the three-objective instance, TwoArch2 reaches the best result.

As a relatively simple question, DTLZ2 is used to test a specific algorithm's diversity ability. It is clear that iTwoArch can achieve the best performance. For the rest of the problems, all the performance of algorithms appears similar except for the five-objective problem. Thus, in order to give more comparison, Figure 1 plots the 3-dimensional radial coordinate visualization (3D-RadVis) to present the same results on five-objective instances [33]. As shown, the proposed iTwoArch and MOEA/D-AM2M have a better performance in terms of both convergence and diversity. MOEA/D can archive a good convergence but struggle to maintain a set of well-distributed individuals. Last, solutions obtained by TwoArch2 fail to get the boundary on the PF, and some of its solutions are far from the optimal front.

The PF of DTLZ3 is exactly the same as DTLZ2, yet it contains too many local optima in its search space. The proposed iTwoArch obtains medium performance in all these instances. TwoArch2 get the worst performance on this problem, its indicator-based convergence mechanism may lose efficiency in higher-dimensional problems. In contrast, MOEAD-AM2M performs better on 8- and 10-objective instances, which means that a suitable mechanism of adaptive weight adjustment may also help in escaping the local optima.

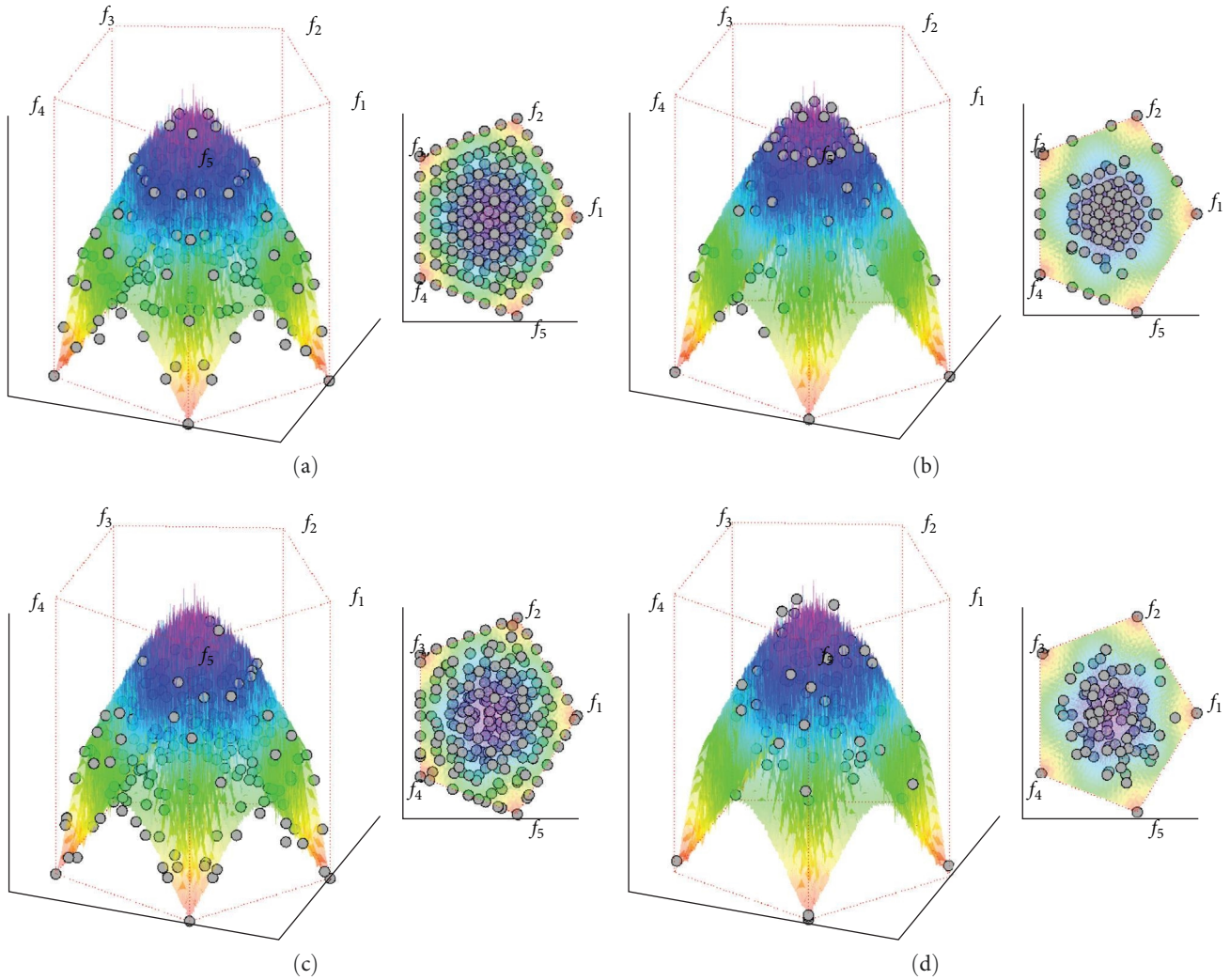

FIGURE 1: 3D-RadVis plots showing obtained results for five-objective DTLZ2 test problem. (a) iTwoArch, (b) MOEA/D, (c) MOEA/D-AM2M, and (d) TwoArch2.

Similar to DTLZ2, the main challenge of DTLZ4 is also on the maintaining diversity of the population. From the statistical results, it can be observed that iTwoArch and NSGA-III have equivalent performances and both algorithms are better than their competitors on 15-objective instance. We can also find that both weight-based and dominance-based algorithms can archive good performance on this problem, which means that only considering DTLZ does not provide sufficient information. Next, other comparison results are devoted to the WFG problems, concerning more analysis of the behavior of our algorithm.

4.2. Performance Comparisons on WFG Problems. WFG test suits pose some powerful challenges for the algorithm by introducing several complexities [28]. Table 3 illustrates the comparison results of iTwoArch with other peer algorithms in terms of HV values.

WFG1 to WFG3 problems have mixed PFs, and it has been empirically proven that the aggregation-based algorithms itself not be suitable for this kind of problem with irregular PF. However, for the WFG1 problem, iTwoArch performs better than the other five algorithms in three- to eight-objective test instances. TwoArch2 is found to archive the best overall performance on 10-objective problems, and it also shows remarkable performance on some instances in WFG2 and WFG3. The diversity mechanism in TwoArch2, which is likely to fail to find boundary solutions in DTLZ, shows relatively competitive performance in WFG1-3 problems. By introducing the adaptive subregion division, MOEA/D-AM2M obtains the best performance on 8- and 10-objective WFG2 problems, which is composed of some disconnected convex. As a connected version of WFG2, WFG3 has a linear and degenerate PF shape. It is worth noting that the aggregation-based methods (iTwoArch, TwoArchA+, MOEA/D, and MOEA/D-AM2M) are significantly outperformed by the NSGA-III in the scope of WFG3 problems. One possible reason for this occurrence is that the mechanism of normalization in NSGA-III can effectively drive the population toward the PF.

Despite WFG4-9 problems having the same hyper-ellipse PF shape, they have different characteristics. For such problems, the observation is similar. The proposed iTwoArch

Table 3: Performance comparison on WFG problems with respect to the average HV values.

Problem	Obj.	iTwoArch	TwoArchA+	TwoArch2	MOEAD	MOEADAM2M	NSGAIII
WFG1	3	**0.926901**	0.856271†	0.867731†	0.911308†	0.915450†	0.819699†
	5	**0.948594**	0.737199†	0.819418†	0.847980†	0.902880†	0.670358†
	8	**0.816376**	0.718838†	0.741628†	0.686077†	0.801972†	0.739105†
	10	0.796913	**0.857521**	0.777469	0.762214†	0.833262	0.742918†
	15	0.711188	0.820765	**0.986308**	0.918149	0.914750	0.952438
WFG2	3	0.952058	0.934914†	**0.960704**	0.926123†	0.948581†	0.956676
	5	**0.994906**	0.993320†	0.993009†	0.957989†	0.987057†	0.988425†
	8	0.987309	0.981483†	0.994552	0.989272	**0.996225**	0.979909†
	10	0.995084	0.990506†	0.995044	0.994259	**0.997953**	0.986114†
	15	0.994858	0.993955	**0.997924**	0.993435	0.951768†	0.995440
WFG3	3	0.695192	0.687666†	**0.716139**	0.705539	0.705136	0.703835
	5	0.660971	0.649774†	0.658714	0.584767†	0.625069†	**0.673299**
	8	0.526521	0.490723†	**0.636992**	0.485826†	0.580795	0.531304
	10	0.505360	0.458569†	0.622333	0.425440†	0.499074	**0.636623**
	15	0.452146	0.324792†	0.604605	0.243627†	0.374232†	**0.605925**
WFG4	3	0.726491	0.715394†	**0.737768**	0.698283†	0.718604†	0.729026
	5	**0.875610**	0.867630†	0.806952†	0.744254†	0.819480†	0.860951†
	8	0.917326	0.906253†	0.820707†	0.822711†	**0.923953**	0.859924†
	10	**0.958477**	0.939936†	0.800069†	0.844913†	0.952985†	0.896477†
	15	0.941565	**0.966244**	0.715636†	0.749120†	0.735066†	0.963551
WFG5	3	0.693701	0.686129†	**0.708057**	0.675991†	0.695819	0.698574
	5	0.833103	0.831710†	0.795315†	0.712879†	**0.837515**	0.835516
	8	**0.870247**	0.869837	0.799887†	0.757722†	0.869645	0.858166†
	10	**0.902923**	0.899994†	0.776144†	0.798187†	0.889022†	0.888981†
	15	0.894864	0.908233	0.670493†	0.783439†	0.636881†	**0.918765**
WFG6	3	0.700455	0.684049†	**0.713969**	0.672674†	0.697281†	0.705634
	5	**0.837602**	0.833408†	0.787224†	0.704609†	0.735872†	0.830531†
	8	0.868866	0.860966†	0.785815†	0.758439†	**0.872055**	0.849389†
	10	**0.901661**	0.886148†	0.749615†	0.772267†	0.887966†	0.880136†
	15	0.865854	0.868941	0.657703†	0.750781†	0.635003†	**0.896825**
WFG7	3	0.726702	0.714129†	**0.739790**	0.704927†	0.717896†	0.730001
	5	**0.873202**	0.868793†	0.825695†	0.736929†	0.810010†	0.872547†
	8	**0.921806**	0.916877†	0.854173†	0.811058†	0.919822†	0.900116†
	10	**0.962508**	0.954990†	0.827919†	0.875988†	0.947103†	0.939205†
	15	0.952157	0.969923	0.723336†	0.750785†	0.693370†	**0.974412**
WFG8	3	0.687804	0.671980†	0.694747	0.667656†	0.677682†	**0.695647**
	5	**0.818049**	0.805905†	0.726794†	0.664497†	0.708814†	0.813213†
	8	0.787465	0.783934	0.685914†	0.676467†	**0.804487**	0.778344†
	10	**0.871625**	0.828460†	0.622675†	0.688680†	0.788674†	0.820759†
	15	0.839984	0.777011†	0.562198†	0.385052†	0.613180†	**0.904617**
WFG9	3	0.706828	0.684670†	**0.721881**	0.675983†	0.691046†	0.704539†
	5	**0.841536**	0.811042†	0.790622†	0.712429†	0.790458†	0.820898†
	8	0.833432	0.813514†	0.804324†	0.777175†	**0.872163**	0.806365†
	10	0.835911	0.872345	0.759983†	0.730805†	**0.879867**	0.859226
	15	0.725454	0.775216	0.604799†	0.540458†	0.633879†	**0.882199**

Note. "†" means that the result is significantly outperformed by iTwoArch.

has a clear advantage over other algorithms. But the performance of algorithms varies not only over problem characteristics but also over the number of objectives. The WFG4 problem, which has a multimodel, is designed for assessing algorithms' ability to escape from the local optima. iTwoArch can win two cases in this problem. TwoArchA+ gives the highest performance on 3- and 50-objective issues. Interestingly, all aggregation-based algorithms are very competitive in this group of problems; only

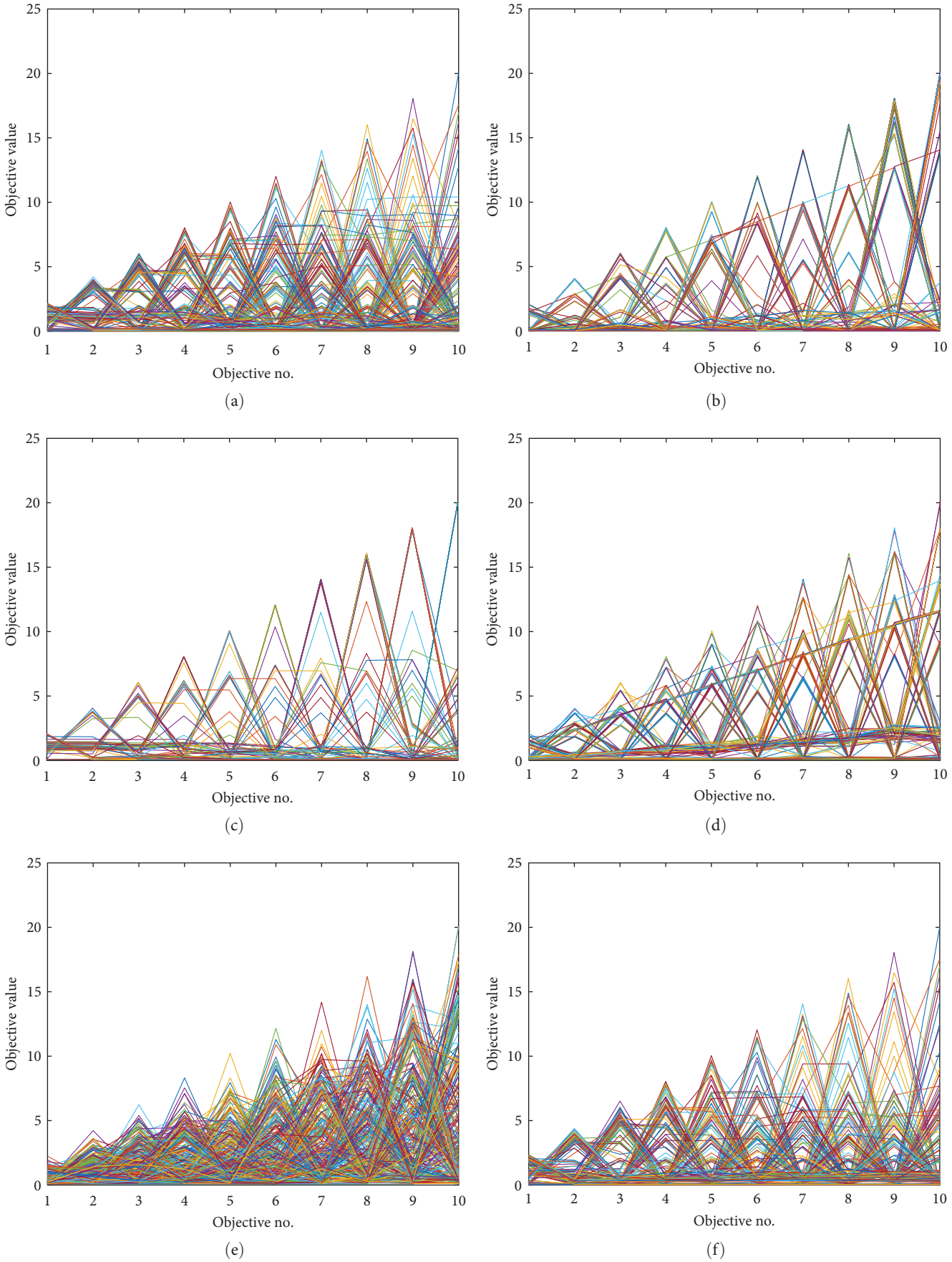

Figure 2: The final solution set of the eight algorithms on the 10-objective WFG6 instance, shown by parallel coordinates. (a) iTwoArch, (b) MOEA/D-AM2M, (c) MOEA/D, (d) NSGAIII, (e) TwoArch2, and (f) TwoArchA+.

Table 4: Summary of the significance test of HV between the proposed algorithm and other algorithms.

		TwoArchA+	TwoArch2	MOEA/D	MOEADAM2M	NSGAIII
iTwoArch vs.	B	46	44	52	45	41
	W	11	17	6	15	19
	E	8	4	7	5	5

"B" ("W") means that the number of instances on which the results proposal are significantly better (worse) than other algorithms. "E" means that the number of instances without detected differences between the results of two algorithms.

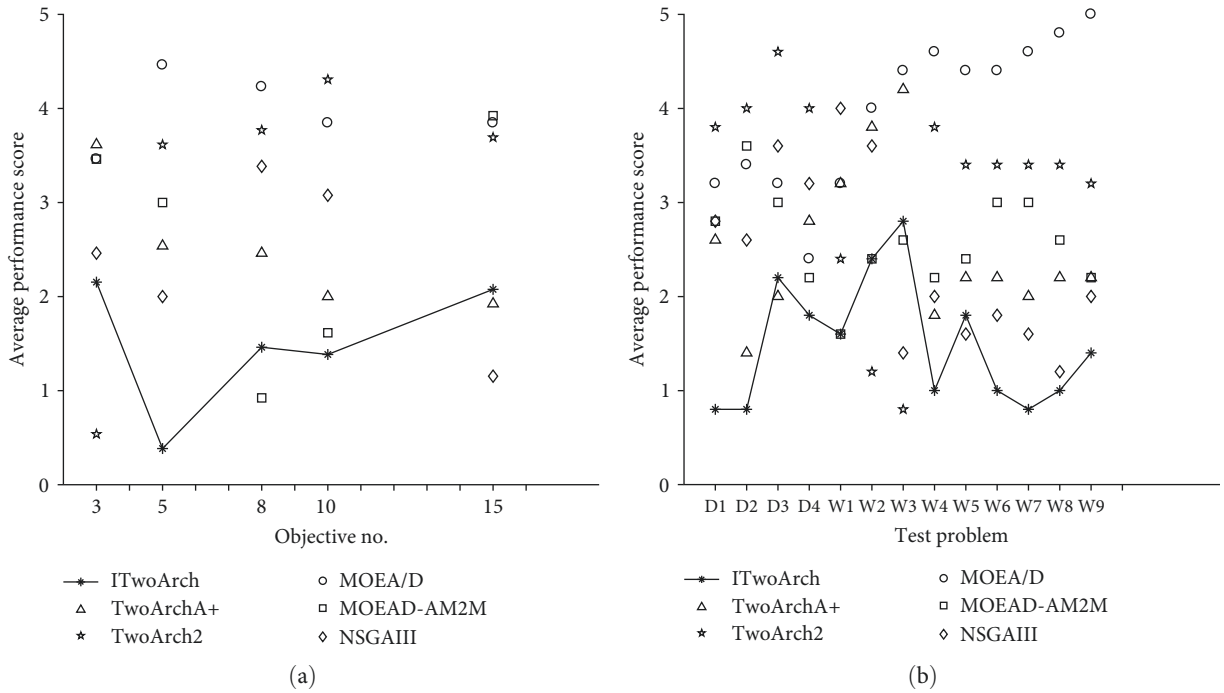

Figure 3: Ranking of average performance score: (a) average performance score over all dimensions for different test cases; (b) average performance score over all test instances for different number of objectives. The values of the proposed ITwoArchA connected by a solid line for clarity.

MOEA/D finds it hard to show a clear advantage. WFG5 is a deceptive problem. The performance of two-archive based methods and NSGA-III demonstrated much better performance than the other algorithms. Conversely, MOEA/D still scale poorly on the high-dimensional objective instances. Similar to the observations before, for WFG6-9 problems, the proposed iTwoArchD averagely wins twice and NSGAIII dose once in 50-objective instances. The deterioration of HV in MOEA/D indicates that applying a single aggregation-based method may struggle to solve high-dimensional WFG problems. Yet, any hybrid strategy may help to alleviate this disadvantage. For example, aided by subregion division and adaptive allocating strategies, MOEA/D-AM2M shows more competitiveness on WFG6-9. The primary reason could be that the subregions in MOEA/D-AM2M protect some solutions to be easily replaced and enhance diversity. Besides, the balance between convergence and diversity could be efficiently archived by incorporating two archives with different purposes. It makes iTwoArch and TwoArch+ behave quite well on WFG problems.

To intuitively investigate the effectiveness of the proposed iTwoArch, the parallel coordinates of the nondominated fronts with median metric value with other compared algorithms are plotted in Figure 2, reflecting the distributions of the solutions obtained by the iTwoArch and five compared algorithms on the 10-objective WFG6. ITwoArchA has a good performance on both convergence and diversity since it can reach the upper and lower boundaries and obtain a better distribution for all 10 objectives.

4.3. Summary of Performance Comparison. In this subsection, we compared the proposed iTwoArch with all the state-of-the-art algorithms mentioned in Section 3.3. Table 4 summarizes the significant test on HV results between the proposed iTwoArch and the peer algorithms. For a specific row on this table, "B"("W") at the second column of a row means the number of instances on which the results of the proposed algorithm are significantly better (or worse) than the compared one, and "E" means the number of results where there is no statistical significance between two compared algorithms.

To better qualify the overall performance of algorithms in the aspect of test problem as well as the number of

objectives, we introduce Conover-Inman [34] procedure to evaluate performance scores. Specially, suppose l algorithms are included for evaluation $\{A_1, A_2, \ldots, A_l\}$. An score for an algorithm A_i is obtained by $P(A_i) = \sum_{j=1, j\neq i}^{l} \delta_{i,j}$, where $\delta_{i,j} = 1$ means that A_j is significantly better than A_i, and 0 otherwise. The score of $P(A_i)$ can reveal how many other algorithms outperform the current one on all test instances. Thus, an algorithm is said to be good if it has a lower score.

At first glance of these summary results (Figure 3), we can get some initial observations: (1) considering the summary of comparison, the proposed iTwoArch gets the most winning times on all the problem instances considered in this study. (2) The proposed algorithm performs best on DTLZ1-2, DTLZ4, WFG1, WFG4, and WFG6-9 problems. (3) iTwoArch performs best on 5- and 10-objective instances and it remains competitive on 8- and 15-objective test problems.

5. Conclusion

This paper aims to propose an improved two-archive evolutionary optimization algorithm, where convergence and diversity are addressed in two separate archives for effectively solving MaOPs. To archive this, we first create two independent archiving procedures using a steady-state manner. The newly created offspring can go through these two archives, which determine whether or not they will survive. We then design an interactive mechanism between these two archives: the index of the best-matched subproblem in CA is directly acquired by updating DA. This interaction may allow producing comparatively higher-quality offspring. A matting selection is carefully designed to balance the convergence and diversity in these two archives. The frequency of replacement in these two archives is counted as an indicator to represent the speed of the evolutionary process. According to this indicator, two individuals for crossover are restricted to distinct situations. Thus, the evolutionary speed of these two archives can be controlled collaboratively through quantitative comparisons on different test problems with five different scales of objective numbers vs. five state-of-the-art peer algorithms. The proposed iTwoArch performs considerably better in addressing MaOPs.

One major future work is to further investigate the proposed iTwoArch in more problems with different characteristics since the performance of MaOPs is also restricted to specific problem features by virtue of the "no free lunch" theorem. In addition, it is also interesting to study how to deal with the problems with irregular PF. This will probably call for the development of new environmental selection mechanisms.

Data Availability

Data supporting this research article are available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is finacially supported by the Natural Science Foundation of China under Grant 40872087.

References

[1] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: an engineering design perspective," in *Evolutionary Multi-Criterion Optimization*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds., vol. 3410 of *Lecture Notes in Computer Science*, pp. 14–32, Springer, Berlin Heidelberg, 2005.

[2] M. W. Mkaouer, M. Kessentini, S. Bechikh, K. Deb, and M. Ó. Cinnéide, "High dimensional search-based software engineering: finding tradeoffs among 15 objectives for automating software refactoring using NSGA-III," in *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation, GECCO '14*, pp. 1263–1270, ACM, New York, NY, USA, 2014.

[3] R. A. Kidanu, M. Cunha, E. Salomons, and A. Ostfeld, "Improving multi-objective optimization methods of water distribution networks," *Water*, vol. 15, no. 14, Article ID 2561, 2023.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] E. Zitzler, M. Laumanns, and T. Lothar, *Spea2: Improving the Strength Pareto Evolutionary Algorithm*, ETH Zurich, Computer Engineering and Networks Laboratory, 2001.

[6] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770–784, 2007.

[7] K. Deb, M. Mohan, and S. Mishra, "Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions." *Evolutionary Computation*, vol. 13, no. 4, pp. 501–525, 2005.

[8] M. Farina and P. Amato, "A fuzzy definition of "optimality" for many-criteria optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 34, no. 3, pp. 315–326, 2004.

[9] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.

[10] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: linear and nonlinear algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 1, pp. 77–99, 2013.

[11] A. L. Jaimes, C. A. Coello Coello, and J. E. Urías Barrientos, "Online objective reduction to deal with many-objective problems," in *Evolutionary Multi-Criterion Optimization*, M. Ehrgott, C. M. Fonseca, X. Gandibleux, J. K. Hao, and M. Sevaux, Eds., vol. 5467 of *Lecture Notes in Computer Science*, pp. 423–437, Springer, Berlin, Heidelberg, 2009.

[12] Y. Yuan, Y.-S. Ong, A. Gupta, and H. Xu, "Objective reduction in many-objective optimization: evolutionary multiobjective approaches and comprehensive analysis," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 189–210, 2018.

[13] J. Bader and E. Zitzler, "Hype: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[14] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature—PPSN VIII*, X. Yao, E. K. Burke, and J. A. Lozano, Eds., vol. 3242 of *Lecture Notes in Computer Science*, pp. 832–842, Springer, Berlin Heidelberg, 2004.

[15] Z. Liang, T. Luo, K. Hu, X. Ma, and Z. Zhu, "An indicator-based many-objective evolutionary algorithm with boundary protection," *IEEE Transactions on Cybernetics*, vol. 51, no. 9, pp. 4553–4566, 2021.

[16] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[17] M. Li and X. Yao, "What weights work for you? adapting weights for any pareto front shape in decomposition-based evolutionary multiobjective optimisation," *Evolutionary Computation*, vol. 28, no. 2, pp. 227–253, 2020.

[18] M. Wu, K. Li, S. Kwong, and Q. Zhang, "Evolutionary many-objective optimization based on adversarial decomposition," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 753–764, 2020.

[19] L. Cai, S. Qu, Y. Yuan, and X. Yao, "A clustering-ranking method for many-objective optimization," *Applied Soft Computing*, vol. 35, pp. 681–694, 2015.

[20] K. Praditwong and X. Yao, "A new multi-objective evolutionary optimisation algorithm: the two-archive algorithm," in *2006 International Conference on Computational Intelligence and Security*, pp. 286–291, IEEE, Guangzhou, China, 2006.

[21] S. Reddy and G. S. Dulikravich, "A self-adapting algorithm for many-objective optimization," *Applied Soft Computing*, vol. 129, Article ID 109484, 2022.

[22] H. Wang, L. Jiao, and X. Yao, "Two_arch2: an improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2015.

[23] L. Cai, S. Qu, and G. Cheng, "Two-archive method for aggregation-based many-objective optimization," *Information Sciences*, vol. 422, pp. 305–317, 2018.

[24] Y. Liu, G. G. Yen, and D. Gong, "A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 660–674, 2019.

[25] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2019.

[26] I. Das and J. E. Dennis, "Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[27] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, A. Abraham, L. Jain, and R. Goldberg, Eds., Advanced Information and Knowledge Processing, pp. 105–145, Springer, London, 2005.

[28] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[29] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[30] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, "Adaptively allocating search effort in challenging many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 433–448, 2018.

[31] J. J. Durillo and A. J. Nebro, "jMetal: a Java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, 2011.

[32] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[33] A. Ibrahim, S. Rahnamayan, M. V. Martin, and K. Deb, "3D-RadVis: visualization of Pareto front in many-objective optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 736–745, IEEE, Vancouver, BC, Canada, 2016.

[34] W. J. Conover, *Practical Nonparametric Statistics*, Vol. 350, John Wiley & Sons, 1998.